UNIVERSITY OF SASKATCHEWAN

Department of Computer Science

CMPT 215.3 MIDTERM EXAMINATION

October 31st, 2000

Total Marks: 50

CLOSED BOOK and CLOSED NOTES
NO CALCULATOR

Time: 75 minutes

Instructions

Read each question carefully and write your answer legibly on the examination paper. **No other paper will be accepted**. You may use the backs of pages for rough work but all final answers must be in the spaces provided. The marks for each question are as indicated. Allocate your time accordingly.

Ensure that your name AND student number are clearly written on the examination paper and that your name is on every page.

Note: a reference table of MIPS instructions is provided at the end of the examination paper.

Question	Marks
1 (5 marks)	
2 (10 marks)	
3 (15 marks)	
4 (20 marks)	
Total	

Name: _____

Student Number: _____

Student Number:_____

1. **General** *(5 marks in total – 1 mark for each part)* Give the technical term that best fits each of the following descriptions or definitions.

   (a) A "law" that quantifies how the system performance enhancement possible with a given improvement is limited by the amount that the improved feature is used.

   (b) Information that tells the linker which words must be changed when an object file is shifted in the address space.

   (c) A style of instruction set architecture in which arithmetic instructions have no explicit operands.

   (d) The laws of this type of algebra are used to simplify logic equations.

   (e) During execution of a program with nested procedure calls, there may be many of these structures on the stack simultaneously.

2. **Computer Performance** *(10 marks in total)*

   (a) (*2 marks*) Consider a program for which 75% of the instructions executed are of a type *A*, and 25% are of a type *B*. Suppose that class *A* instructions each require 2 clock cycles on average. If the overall CPI of the program is 2.5, how many clock cycles must be required by each class *B* instruction on average?

(b) (*6 marks*) In each of the following parts, state which ones of the three factors determining CPU execution time (number of machine language instructions executed, clock cycle time, CPI) may change, when the indicated system change is made (and all else is held constant).

(i) a new compiler is used

(ii) a different implementation of the (same) instruction set architecture is used

(c) (*2 marks*) Suppose that the MIPS rating for a particular program is 500. If the CPI is 2.0, what is the clock period length in nanoseconds? (Recall that a nanosecond is 1.0×10^{-9} seconds.)

3. **Arithmetic** *(15 marks in total)*

(a) *(3 marks)* Perform the following conversions.

(i) Convert 28_{10} to a base 3 number.

(ii) Convert -23_{10} to a 6-bit 2's complement binary number.

(iii) Convert 7206_8 to a base 2 number.

(b) *(4 marks)* Suppose we wish to find the sum of three numbers X, Y, Z at some point in a computer program.

(i) If the numbers are floating point numbers, can the order in which we add them (e.g., (X+Y)+Z vs. X+(Y+Z)) make any difference (i.e., have any impact on the outcome)? Explain your answer.

(ii) Repeat part (i), but now assuming that the numbers are signed integers that we add using the MIPS "add" instruction.

(iii) Repeat part (i), but now assuming that the numbers are signed integers that we add using the MIPS "addu" instruction.

(c) *(2 marks)* Recall that in the IEEE 754 floating point standard, single precision floating point numbers have a 1 bit sign field, followed by an 8 bit exponent field (in biased notation with a bias of 127), followed by a 23 bit significant field. Give the representation of –3.5625 in IEEE 754 single precision format.

(d) *(4 marks)* In the ALU design discussed in class, the ALU accepted two 32 bit inputs, and produced a 32 bit result. The operations supported included addition and subtraction. It is desirable to have an additional 1 bit output from the adder that equals 1 if signed overflow occurs during an addition or subtraction operation, and 0 otherwise. Give a **truth table** for such a 1 bit output, as a logic function of the high order bits of the operands (a_{31} and b_{31}), the high order bit of the result ($result_{31}$), and "*Binvert*" (which equals 1 if the operation is a-b, and equals zero if the operation is a+b). Then, using your truth table, derive a **logic equation** in sum-of-products form. (You **don't** need to simplify it.)

(e) *(2 marks)* If Booth's algorithm was used to multiply the 8-bit signed binary number 01101111 by the 8-bit signed binary number 11010100, how many times would the multiplicand have to be added or subtracted?

4. **Machine and Assembly Language** *(20 marks in total)*

(a) *(5 marks)* List the five MIPS addressing modes, giving (and briefly describing how it works) an example of each.

(b) *(3 marks)* Consider the following code fragment.

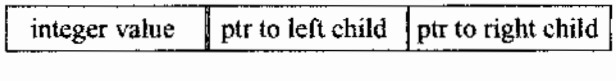
```
        .data
A:      .word 2
        .word 4
        .word 6
        .word 8
        .word 10

        .text
main:   la    $t0, A
        lw    $t1, 4($t0)
        sub   $t2, $t1, $t1
        li    $s0, 10
loop:   addi  $t2, $t2, 1
        add   $s0, $t1, $s0
        bne   $t2, $t1, loop
```

(i) Following execution of the above code, what are the contents of $s0?

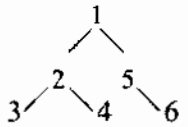
(ii) What would happen if we ran the above code with the "lw" instruction replaced by "lw $t1, 3($t0)"?

Student Number:_____

(c) *(12 marks)* Write a **recursive** MIPS procedure that prints the integer values stored in a binary tree, using a "preorder" traversal. Each node of the tree is implemented with three consecutive words of memory, as shown below on the left. On the right is an example tree, and the output that your procedure should give for this tree.



address x address x+4 address x+8



output integers 1,2,3,4,5,6 (in that order)

Each child pointer gives the memory address of the first word of the corresponding child node, with a memory address of zero indicating that there is no such child node. To implement a preorder traversal, you print the root node value, then the values from the left subtree (using a recursive call), then the values from the right subtree (using a recursive call). Your procedure should take a single argument, the address of the root node of the tree (or subtree), in $a0. Recall that the syscall code for printing an integer is "1".

The End